

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re U.S. Patent Application)	<i>I hereby certify that this paper is being deposited with the United States Postal Service as EXPRESS MAIL in an envelope addressed to:</i>
Applicant: Manabu Matsuyama)	<i>Mail Stop PATENT APPLICATION, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on this date.</i>
Serial No.)	<u>March 29, 2004</u>
)	Date Express Mail Label No.: EV032736264US
Filed: March 29, 2004)	
For: METHOD, APPARATUS AND)	
COMPUTER PROGRAM FOR)	
COMPILING PROGRAM USING)	
STATISTICAL INFORMATION)	
ON VARIABLES, AND RECORDING)	
MEDIUM THEREFOR)	
Art Unit:)	

CLAIM FOR PRIORITY

Mail Stop PATENT APPLICATION
 Commissioner for Patents
 P.O. Box 1450
 Alexandria, VA 22313-1450

Dear Sir:

Applicants claim foreign priority benefits under 35 U.S.C. § 119 on the basis of the foreign application identified below:

Japanese Patent Application No. 2003-301039, filed August 26, 2003.

A certified copy of the priority document is enclosed.

Respectfully submitted,

GREER, BURNS & CRAIN, LTD.

By

James K. Folker
 Registration No. 37,538

Customer No. 24978

March 29, 2004
 300 South Wacker Drive - Suite 2500
 Chicago, Illinois 60606
 Phone: (312) 360-0080
 Fax: (312) 360-9315

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日 2 0 0 3 年 8 月 2 6 日
Date of Application:

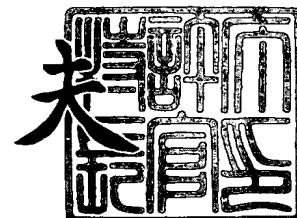
出 願 番 号 特 願 2 0 0 3 - 3 0 1 0 3 9
Application Number:
[ST. 10/C]: [J P 2 0 0 3 - 3 0 1 0 3 9]

出 願 人 富 士 通 株 式 会 社
Applicant(s):

2 0 0 4 年 2 月 2 5 日

特許庁長官
Commissioner,
Japan Patent Office

今 井 康 夫



【書類名】 特許願
【整理番号】 0350624
【提出日】 平成15年 8月26日
【あて先】 特許庁長官殿
【国際特許分類】 G06F 9/45
【発明者】
 【住所又は居所】 神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号 富士通株式会社
 内
 【氏名】 松山 学
【特許出願人】
 【識別番号】 000005223
 【氏名又は名称】 富士通株式会社
【代理人】
 【識別番号】 100087848
 【弁理士】
 【氏名又は名称】 小笠原 吉義
 【電話番号】 03-3807-1151
【選任した代理人】
 【識別番号】 100083297
 【弁理士】
 【氏名又は名称】 山谷 皓榮
【手数料の表示】
 【予納台帳番号】 012586
 【納付金額】 21,000円
【提出物件の目録】
 【物件名】 特許請求の範囲 1
 【物件名】 明細書 1
 【物件名】 図面 1
 【物件名】 要約書 1
 【包括委任状番号】 0203885

【書類名】 特許請求の範囲**【請求項 1】**

プログラム言語で記述されたプログラムを翻訳してコンピュータが実行可能な目的プログラムを生成するコンパイル処理を、コンピュータに実行させるためのコンパイル処理プログラムにおいて、

第 1 回目のプログラム翻訳時に、前記プログラムの実行中に手続き内の変数に対して与えられる値と前記値が与えられる回数との組を最大 2 組までレコードに記録する統計処理を行う命令列と、前記プログラムの実行終了後に前記レコードに記憶された情報を一次プロファイル情報として出力する一次プロファイル出力処理を行う命令列とを、前記プログラムに付加する処理と、

前記第 1 回目のプログラム実行の終了後に、出力された前記一次プロファイル情報を記憶部に記憶する処理と、

前記一次プロファイル情報を読み込み、前記一次プロファイル情報に記憶された前記値を前記レコードに設定する処理と、

第 2 回目のプログラム翻訳時に、前記レコードに設定された値の出現回数と前記値を変数に与える処理を含む手続きの実行回数とを前記レコードに記録する統計検証処理を行う命令列と、前記プログラムの実行終了後に前記レコードに記憶された前記出現回数および前記実行回数とを最終プロファイル情報として出力する最終プロファイル出力処理を行う命令列とを、前記プログラムに付加する処理と、

前記第 2 回目のプログラム実行の終了後に、出力された前記最終プロファイル情報を記憶部に記憶する処理とを、

コンピュータに実行させる

ことを特徴とする変数の統計処理を行うコンパイル処理プログラム。

【請求項 2】

さらに、前記最終プロファイル情報を読み込む処理と、

第 3 回目のプログラム翻訳時に、前記最終プロファイル情報から取得した前記値の出現回数と前記値を変数に与える処理を含む手続きの実行回数にもとづいて前記値の出現頻度を算出する処理と、

前記出現頻度が 50% 以上である場合に、前記値により予測される動作を行う処理に最適化する処理とを、

コンピュータに実行させる

ことを特徴とする請求項 1 記載の変数の統計処理を行うコンパイル処理プログラム。

【請求項 3】

プログラム言語で記述されたプログラムを翻訳してコンピュータが実行可能な目的プログラムを生成するコンパイル処理を、コンピュータに実行させるためのコンパイル処理プログラムを記録する記録媒体であって、

第 1 回目のプログラム翻訳時に、前記プログラムの実行中に手続き内の変数に対して与えられる値と前記値が与えられる回数との組を最大 2 組までレコードに記録する統計処理を行う命令列と、前記プログラムの実行終了後に前記レコードに記憶された情報を一次プロファイル情報として出力する一次プロファイル出力処理を行う命令列とを、前記プログラムに付加する処理と、

前記第 1 回目のプログラム実行の終了後に、出力された前記一次プロファイル情報を記憶部に記憶する処理と、

前記一次プロファイル情報を読み込み、前記一次プロファイル情報に記憶された前記値を前記レコードに設定する処理と、

第 2 回目のプログラム翻訳時に、前記レコードに設定された値の出現回数と前記値を変数に与える処理を含む手続きの実行回数とを前記レコードに記録する統計検証処理を行う命令列と、前記プログラムの実行終了後に前記レコードに記憶された前記出現回数および前記実行回数とを最終プロファイル情報として出力する最終プロファイル出力処理を行う命令列とを、前記プログラムに付加する処理と、

前記第2回目のプログラム実行の終了後に、出力された前記最終プロファイル情報を記憶部に記憶する処理とを、

コンピュータに実行させるプログラムを記録する

ことを特徴とする変数の統計処理を行うコンパイル処理プログラム記録媒体。

【請求項4】

プログラム言語で記述されたプログラムを翻訳してコンピュータが実行可能な目的プログラムを生成するコンパイル処理方法において、

第1回目のプログラム翻訳時に、前記プログラムの実行中に手続き内の変数に対して与えられる値と前記値が与えられる回数との組を最大2組までレコードに記録する統計処理を行う命令列と、前記プログラムの実行終了後に前記レコードに記憶された情報を一次プロファイル情報として出力する一次プロファイル出力処理を行う命令列とを、前記プログラムに付加し、

前記第1回目のプログラム実行の終了後に、出力された前記一次プロファイル情報を記憶部に記憶し、

前記一次プロファイル情報を読み込み、前記一次プロファイル情報に記憶された前記値を前記レコードに設定し、

第2回目のプログラム翻訳時に、前記レコードに設定された値の出現回数と前記値を変数に与える処理を含む手続きの実行回数とを前記レコードに記録する統計検証処理を行う命令列と、前記プログラムの実行終了後に前記レコードに記憶された前記出現回数および前記実行回数とを最終プロファイル情報として出力する最終プロファイル出力処理を行う命令列とを、前記プログラムに付加し、

前記第2回目のプログラム実行の終了後に、出力された前記最終プロファイル情報を記憶部に記憶する、

ことを特徴とする変数の統計処理を行うコンパイル処理方法。

【請求項5】

プログラム言語で記述されたプログラムを翻訳してコンピュータが実行可能な目的プログラムを生成するコンパイル処理装置において、

プロファイル情報を記憶する記憶部と、

第1回目のプログラム翻訳時に、前記プログラムの実行中に手続き内の変数に対して与えられる値と前記値が与えられる回数との組を最大2組までレコードに記録する統計処理を行う命令列と、前記プログラムの実行終了後に前記レコードに記憶された情報を一次プロファイル情報として出力する一次プロファイル出力処理を行う命令列とを、前記プログラムに付加する処理手段と、

前記第1回目のプログラム実行の終了後に、出力された前記一次プロファイル情報を記憶部に記憶する処理手段と、

前記一次プロファイル情報を読み込み、前記一次プロファイル情報に記憶された前記値を前記レコードに設定する処理手段と、

第2回目のプログラム翻訳時に、前記レコードに設定された値の出現回数と前記値を変数に与える処理を含む手続きの実行回数とを前記レコードに記録する統計検証処理を行う命令列と、前記プログラムの実行終了後に前記レコードに記憶された前記出現回数および前記実行回数とを最終プロファイル情報として出力する最終プロファイル出力処理を行う命令列とを、前記プログラムに付加する処理手段と、

前記第2回目のプログラム実行の終了後に、出力された前記最終プロファイル情報を記憶部に記憶する処理手段とを、備える

ことを特徴とする変数の統計処理を行うコンパイル処理装置。

【書類名】明細書

【発明の名称】変数の統計処理を行うコンパイル処理プログラム、およびその記録媒体、およびその処理方法ならびにその処理装置

【技術分野】

【0001】

本発明は、プログラム言語で記述された原始プログラムを翻訳して目的プログラムを生成するコンパイラ処理に関する。より詳しくは、本発明は、翻訳対象のプログラムの手続き中の変数に対して実行時に与えられる実際の値の統計情報を収集し、その統計情報を用いて手続きの最適化を行うコンパイラ処理に関する。

【背景技術】

【0002】

コンピュータにおけるプログラムのコンパイル処理において、原始プログラムを翻訳し生成した目的プログラムの実行中に、その手続き中の変数に実際に与えられた値とその頻度とを統計情報（プロファイル情報）として記録し、この記録したプロファイル情報を次回以降の翻訳処理において手続きを最適化するために利用する方法がある（例えば、特許文献1参照）。

【0003】

特許文献1「コンパイラとプログラム最適化方法およびその処理プログラムを記録した記録媒体」に記載されている手法では、各手続きのプロファイルの対象となる変数の1つ1つにプロファイルレコード（以後単に「レコード」と呼ぶ）が用意される。そして、目的プログラムの実行時にその変数に与えられる値の種別の1つ1つに対し、値と与えられた回数との組を格納するためのレコード中のデータ構造（以後「値リスト」と呼ぶ）を、単に複数個用意することで、実行時に得られる値を頻度の高い順に保存できるとしているが、特許文献1には、そのように作用する処理の詳細については述べられていない。

【0004】

しかし、特許文献1に記載された説明に従う限りでは次の二通りが考えられる。

【0005】

1) 従来手法A：先着順に登録

対象プログラムの手続きの実行時に、手続きが呼び出された順に変数に与えられた値を登録してゆき、値の種別がレコードの値リストの個数となった場合には、その時点以降に現れた新しい種別の値は登録しないものとする。

【0006】

例えば、`sub(x)` という1個の引数 `x` を取る手続きの入りで変数 `x` に与えられた実際の値を記録することを考える。この手続きが10回呼び出されたときに、変数 `x` に与えられた値が次のようだったとする。

【0007】

`x = {4, 4, 4, 7, 7, 10, 10, 3, 6, 10}`

ここで、値リストの大きさが3であった場合に、3種類目の値“10”が与えられた時点でレコードの全ての値リストには値が格納されているため、それ以降に出現する新しい種類の値（“3”と“6”）は登録しない。したがって、プログラムの終了時点での値リストの状態は以下のようになる。

【0008】

`{値, 回数} = {4, 3}, {7, 2}, {10, 3}`

2) 従来手法B：レコードの値リストが一杯になった時点で最小出現回数の値を切り捨て

対象プログラムの手続きの実行時に、手続きが呼び出された順に変数に与えられた値を登録してゆき、レコードの全ての値リストに値が格納されている場合に、その時点以降に新しい種別の値が現れた場合は、その時点で最も出現回数の少ない値を値リストから追い出し、新しい値を登録する。最も出現回数の少ない値が複数あった場合にはその中から適当に1つの値を選ぶ。

【0009】

例えば、上記の例と同じ手続きが10回呼び出されたときに、変数xに与えられた値が次のようだったとする。

【0010】

x = {4, 4, 4, 7, 7, 10, 3, 6, 8, 8}

このとき、値リストの大きさが3であった場合に、6回目の呼び出しの時点での値リストの状態は以下のようになる。

【0011】

{値, 回数} = {4, 3}, {7, 2}, {10, 1}

続いて、7回目に新しい値“3”と共に呼び出されたとき、全ての値リストはすでに値が格納されているので、この方法では最も出現回数の少ない値“10”を値リストから追い出して、新しい値“3”を登録する。結果は以下のようになる。

【0012】

{値, 回数} = {4, 3}, {7, 2}, {3, 1}

さらに8回目の呼び出しでは、同様に値“3”が追い出されて代わりに値“6”が登録され、9回目の呼び出しでは同様に値“6”が追い出されて値“8”が登録される。この結果、プログラムの終了時点での値リストの状態は以下のようになる。

【0013】

{値, 回数} = {4, 3}, {7, 2}, {8, 2}

【特許文献1】特開平11-149380号公報

【発明の開示】

【発明が解決しようとする課題】

【0014】

上記の2通りの従来手法には、それぞれ以下のような問題点がある。

【0015】

1) 従来手法Aの問題点

上記の処理例において、以下のような順序で値が与えられて手続きが呼ばれた場合に、レコードの値リストには、最も出現回数の高い値が保存されていない可能性がある。

【0016】

x = {1, 2, 3, 4, 4, 5, 5, 6, 6, 6}

この場合であれば、正しくは、以下のような値と回数との組が値リストに保存されていなければならない。

【0017】

{値, 回数} = {6, 3}, {4, 2}, {5, 2}

しかし、実際に保存される組は以下のとおりとなる。

【0018】

{値, 回数} = {1, 1}, {2, 1}, {3, 1}

2) 従来手法Bの問題点

上記の処理例でいえば、以下のような順序で値が与えられて手続きが呼ばれた場合に、値リストには最も出現回数の高い値が保存されていない可能性がある。

【0019】

x = {1, 3, 3, 5, 5, 7, 7, 1, 6, 1}

この場合であれば、正しくは、以下のような値と回数との組が値リストに保存されていなければならない。

【0020】

{値, 回数} = {1, 3}, {3, 2}, {7, 2}

または {1, 3}, {3, 2}, {5, 2}

しかし、実際に保存される組は以下のとおりとなる。

【0021】

{値, 回数} = {3, 2}, {5, 2}, {1, 1}

または {7, 2}, {5, 2}, {1, 1}

または {7, 2}, {3, 2}, {1, 1}

なお、実際にはどのような値と回数との組が値リストに保存されるかは、値リストが一杯になった時点でどの組を捨てるかによって変わることになる。

【0022】

いずれの従来方法においても、手続きの実行時に得られる変数に与えられる値の種別が値リストの個数以上である場合に、値リストに保存された値と回数との組は信頼できないものとなる可能性があることがわかる。

【0023】

特許文献1では明示的に言及されていないが、値リストの大きさは翻訳時に決定しなければならない。もし翻訳時に値リストの大きさを決定せずに実行時に動的に値リスト用の記憶領域を確保するとすれば、目的プログラム自身が動的に記憶領域を確保する処理を持つ場合に、その処理と干渉して悪影響を及ぼす可能性がある。そのため、例えば、J A V A (米国 Sun Microsystems, Inc. の登録商標) の J I T (J u s t I n T i m e) コンパイラにおける仮想マシンのように、システム側で適切な記憶管理を行える場合を除いて、一般的に安全には実行できない。

【0024】

したがって、翻訳前に、値リストの大きさを、変数に与えられる値の種別数より十分に大きくして決定しておくことは實際上不可能である。

【0025】

本発明の目的は、上記の問題点に鑑みて、プログラムのコンパイル処理において、翻訳対象のプログラムの実行中に、その手続きの変数に与えられる値であって50%以上の出現頻度のものについてのプロファイル情報を確実に収集できるコンパイル処理をコンピュータに実行させるための処理プログラムを提供することである。

【0026】

さらに、上記の処理プログラムにより収集したプロファイル情報を利用して最適化を行えるコンパイル処理をコンピュータに実行させるための処理プログラムを提供することである。

【0027】

さらに、上記の処理プログラムを記録した記録媒体、および、上記処理プログラムで実行される処理を実現できる処理方法ならびに処理装置を提供することである。

【課題を解決するための手段】

【0028】

上記の問題点を解決するため、本発明は、プログラム言語で記述されたプログラムを翻訳してコンピュータが実行可能な目的プログラムを生成するコンパイル処理を、コンピュータに実行させるためのコンパイル処理プログラムにおいて、以下の処理を行うものである。

【0029】

本発明は、第1回目のプログラム翻訳時に、前記プログラムの実行中に手続き内の変数に対して与えられる値と前記値が与えられる回数との組を最大2組までレコードに記録する統計処理を行う命令列と、前記プログラムの実行終了後に前記レコードに記憶された情報を一次プロファイル情報として出力する一次プロファイル出力処理を行う命令列とを、前記プログラムに付加する処理と、前記第1回目のプログラム実行の終了後に、出力された前記一次プロファイル情報を記憶部に記憶する処理と、前記一次プロファイル情報を読み込み、前記一次プロファイル情報に記憶された前記値を前記レコードに設定する処理と、第2回目のプログラム翻訳時に、前記レコードに設定された値の出現回数と前記値を変数に与える処理を含む手続きの実行回数とを前記レコードに記録する統計検証処理を行う命令列と、前記プログラムの実行終了後に前記レコードに記憶された前記出現回数および前記実行回数とを最終プロファイル情報として出力する最終プロファイル出力処理を行う命令列とを、前記プログラムに付加する処理と、前記第2回目のプログラム実行の終了後

に、出力された前記最終プロファイル情報を記憶部に記憶する処理とを、コンピュータに実行させるものである。

【0030】

また、本発明は、上記の処理プログラムを記録した記録媒体、または上記の処理を実現する処理方法ならびに処理装置である。

【0031】

本発明は、原始プログラムから生成された目的プログラムの実行中に、その手続きの変数に与えられる値であって50%以上の出現頻度のものについてのプロファイル情報を収集するために、コンパイル処理において以下の2つの処理を行う。

【0032】

(1) 一次プロファイル処理

本発明では、まず、翻訳対象のプログラムの手続き中の変数に与えられる値のうち出現頻度が50%以上となりうる値を記録する処理(統計処理)を、翻訳対象のプログラムの適切な位置に挿入する処理を行う。ここで、統計処理の結果を記録するレコードとして、値リストの大きさを2に固定したレコードを用意する。

【0033】

そして、翻訳対象プログラムの実行中に、変数に対する出現頻度が50%以上になりうる値をレコードに記録し、レコードに記録された2つの値を一次プロファイル情報として出力する。

【0034】

なお、出現頻度が50%以上になりうる値を記録する統計処理は、従来手法Bと同様の手法とする。すなわち、2つの値レコードに既に値が格納されている場合には、その時点以降に新しい種別の値が現れたときは、その時点で出現回数の少ない値を値リストから追い出し、新しい値を格納する。

【0035】

ここでは、後の統計検出処理において出現頻度が50%以上になりうる値が記録されるようにするだけでよく、その出現回数を正確に記録する必要はない。例えば、上記の手続き $sub(x)$ の例で、仮引数 x に与えられた値が以下のようなものだったとする。

【0036】

$x = \{1, 2, 2, 3, 1, 3, 3, 1, 1, 1\}$

この場合に、実際に保存される組は以下のようになり、出現頻度50%の値“1”が記録される。

【0037】

$\{値, 回数\} = \{2, 2\}, \{1, 3\}$ または $\{3, 2\}, \{1, 3\}$

つまり、ある特定の値の出現頻度が50%以上になりうるものであるならば、その値は、実行中の出現間隔が一時的にまばらな期間があるために一旦値リストから追い出されたとしても、実行中のいつかの時点で十分な頻度で出現するために必ず値リストに再格納されるので、大きさ2の値リストを持つレコードに記録されることになるからである。

【0038】

しかし、このような統計処理だけでは、以下のような2つの問題点がある。

【0039】

1) 値の出現回数は正しくない、すなわち実際より低く見積もられることがある。

【0040】

2) 値リストには出現頻度が50%未満の値も登録されることがある。

【0041】

そこで、最終プロファイル処理により、上記の統計処理の処理結果を修正して問題点を解決する。

【0042】

(2) 最終プロファイル処理

本発明では、さらに、1) 変数に値を与える処理を含む基本ブロックの実行回数を計測

する処理と、2) 変数に与えられた値が前記の統計処理で得られたものであった場合に、その値の出現回数を計測する処理との2つの処理(以下、2つの処理を「統計検証処理」とする)を、翻訳対象のプログラムの適切な位置に挿入する処理を行う。

【0043】

なお、基本ブロックとは、1個以上の命令からなる命令列であって、該命令列の末尾以外に分岐命令がなく、該命令列の先頭以外に分岐命令の分岐先となる命令がない命令列である。

【0044】

ここで、統計検証処理の結果を記録するレコードとして、統計処理で用いたレコードと同様に値リストの大きさが2に固定されているレコードを用意し、一次プロファイル情報として記憶されている2つの値をレコードの値リストに設定する。

【0045】

そして、一次プロファイル処理においてプログラムを実行したときと同じ条件のもとでプログラムを実行し、手続き中の変数に与えられた値が、レコードの値リストに設定された値(ここでは、「2と1のいずれか」または「3と1のいずれか」と同じである場合のみ、その値の出現回数をカウント(1つ増分)していく。また、手続き呼出し自体の呼出し回数を記録するレコードを設け、手続きの実行回数をカウント(1つ増分)する。そして、プログラムの実行終了後に、レコードの記録から最終プロファイル情報を出力する。

【0046】

このように、最終プロファイル処理を経た後は、任意の変数に対し、

- 1) 出現頻度が50%以上の値があるかないか、
- 2) あるとすれば何(値)であるか、

についての統計情報(最終プロファイル情報)が取得できる。

【0047】

したがって、以降の翻訳処理での最適化において、この最終プロファイル情報を使用することにより目的プログラムの高速化を図ることができる。

【0048】

以降の翻訳処理において、最終プロファイル情報を参照して最適化を行い、目的プログラムを生成する。例えば、最終プロファイル情報の手続きの実行回数と値の出現回数とをもとに2つの値の出現頻度をそれぞれ算出し、出現頻度が50%以上である値について、その値が与えられる変数を含む手続きを、予測した動作を行うように特殊化する最適化を行う。

【発明の効果】

【0049】

本発明によれば、原始プログラムを翻訳して目的プログラムを生成するコンパイラの処理において、翻訳対象のプログラムの手続き中の変数に実際に与えられる値とその頻度とをプロファイル情報として記録する場合に、手続き中の変数に与えられる値の出現回数を誤りなく見積もることができ、特に、出現頻度が50%を超える値を確実に記録したプロファイル情報を出力することができる。

【0050】

よって、実行時に手続き中の変数に対して同一の値が高頻度で与えられるようなプログラムを翻訳処理する場合に、変数に与えられる特定の値により予測される動作を行うような処理に特殊化する最適化を施すことができ、一層高速化、効率化した目的プログラムを生成することができる。

【0051】

本発明は、プログラムのコンパイルを行う一般的なコンパイル処理装置に利用することが可能であるが、変数に特定の値が参照されやすいようなプログラムについて、高速化な目的プログラムを生成するためのコンパイル処理装置に利用することができる。

【0052】

例えば、組込み機器制御用のプログラムの最適化かつ高速化な目的プログラムを生成するためのコンパイル処理装置に利用することができる。

【0053】

具体的には、自動車のエンジン、ブレーキなどの制御用機器では、ある1つの機器が複数の車種やグレードに採用されることが多い。この場合に、車種やグレードによってエンジン形式、車載装置、車体重量などが異なることが多く、制御対象の相違が機器制御用プログラム内のパラメータや外部センサから入力されるデータの典型値に影響することがある。しかし、このようなデータは事前に計算で求めることが困難であり、試走行などにより取得したデータから最適値を求めてプログラムに反映させる作業は、人手により行っていた。

【0054】

したがって、本発明をコンパイル処理装置に適用することにより、このような最適値のプログラムへの反映を自動化し、より最適化、高速化した目的プログラムを取得できるため、プログラム開発時における負担の軽減を図ることができる。

【発明を実施するための最良の形態】

【0055】

本発明の最良の実施の形態として、本発明を、原始プログラムの翻訳処理において、翻訳時に、実行中の手続きの変数に実際に与えられる値に関するプロファイル情報を記録する処理を付加し、目的プログラムの実行中にそのプロファイル情報を収集・出力し、出力されたプロファイル情報を利用して最適化を行うコンパイル処理装置において実施した。

【実施例】

【0056】

図1は、本発明の一実施例における構成例を示す図である。

【0057】

コンパイル処理装置1は、解析処理部11、一次プロファイル処理部12、一次プロファイル読み込み部13、最終プロファイル処理部14、最終プロファイル読み込み部15、最適化部16、および目的プログラム生成部17を備える。

【0058】

解析処理部11は、翻訳対象の原始プログラム4を読み込み、構文やデータを解析し、原始プログラムから中間形式データを生成する処理手段である。

【0059】

一次プロファイル処理部12は、目的プログラム5の実行中に手続き中の変数に与えられる値のうち出現頻度が50%以上となりうるものをレコード6に記録する統計処理を行う命令列と、実行終了後にレコード6の内容をまとめて一次プロファイル情報31としてファイル出力する処理を行う命令列とを、中間形式データの所定の位置に埋め込む処理手段である。

【0060】

一次プロファイル読み込み部13は、一次プロファイル情報31を読み込む処理手段である。

【0061】

最終プロファイル処理部14は、目的プログラム5の変数に値を与える処理を含む基本ブロックの実行回数と、変数に与えられる値であって一次プロファイル情報31に記憶されたものの出現回数とを計測する統計検証処理を行う命令列と、実行終了後にレコード6の内容をまとめて最終プロファイル情報32としてファイル出力する処理を行う命令列とを、中間形式データの所定の位置に埋め込む処理手段である。

【0062】

最終プロファイル読み込み部15は、最終プロファイル情報32を読み込み、最終プロファイル情報32に設定された変数に与えられる2つの値それぞれについて値の出現頻度を求める処理手段である。

【0063】

最適化部 16 は、最終プロファイル情報 32 を参照して、実行中に手続き中の変数に与えられる値のうち出現頻度が 50% 以上のものに対し、その値により予測される動作を行うような最適化を行う処理手段である。

【0064】

目的プログラム生成部 17 は、中間形式データから目的プログラム 5 を生成する処理手段である。

【0065】

目的プログラム 5 は、実行終了後に、一次プロファイル情報 31 または最終プロファイル情報 32 をファイル出力する。

【0066】

図 2 に、本発明の処理フローを示す。

【0067】

コンパイル処理装置 1 は、原始プログラム 4 に対して 3 回の翻訳処理を行い、最後の翻訳処理において最適化した目的プログラム 5 を生成する。

【0068】

まず、第 1 回目の翻訳処理を行うため、コンパイル処理装置 1 は、一時プロファイル処理部 12 を実施する指示を受けて起動される。

【0069】

コンパイル処理装置 1 は、解析処理部 11 により、原始プログラム 4 を読み込み、必要な構文解析やデータ解析などを行い、原始プログラム 4 を中間形式データに変換する。その際に原始プログラム 4 内の全ての基本ブロックと変数に固有の番号を振る（ステップ S101）。

【0070】

次に、一次プロファイル処理部 12 により、プロファイル情報を取るよう指定した変数の 1 つ 1 つに対し、以下のような処理を行う（ステップ S102）。

【0071】

まず、一次プロファイル処理部 12 は、中間形式データのデータ部分にその変数用のレコード 6 を用意する。そして、レコード 6 の基本ブロック番号欄と変数番号欄を設定し、中間形式データ内のその変数の定義点の直後に、統計処理を行う命令列を挿入する。

【0072】

図 3 (A) に、レコード 6 の構成例を示す。レコード 6 は、基本ブロックの固有の番号が設定される基本ブロック番号、基本ブロックが実行された回数が設定される基本ブロック実行カウンタ、変数の固有の番号が設定される変数番号、プログラムの実行中に基本ブロック中のその変数に与えられる値が設定される値 V および値 V の出現回数が設定されるカウンタ N の 2 つの項目を持つ 2 つの値リスト A および値リスト B で構成される。図 3 (B) は、図 3 (A) のレコード 6 のデータ構造体の C 言語による記述例である。

【0073】

その後、目的プログラム生成部 17 により、中間形式データから目的プログラム 5 を生成する（ステップ S103）。

【0074】

さらに、人手により、その目的プログラム 5 を仮に実行する（ステップ S104）。目的プログラム 5 の実行中、ステップ S102 の処理で挿入された命令列により、以下のような統計処理が行われる。

【0075】

図 4 に、統計処理の流れを示す。

【0076】

まず、目的プログラム 5 の実行中に変数に与えられた値を取得し（ステップ S20）、取得した値がレコード 6 の値リスト A にあるかどうかを判定する（ステップ S21）。

【0077】

取得した値がレコード 6 の値リスト A の値 V a と同じであれば、カウンタ N a を 1 つ増

やす (ステップ S 2 2)。

【0078】

一方、取得した値が値リスト A になれば、その値がレコード 6 の値リスト B にあるかどうかを判定する (ステップ S 2 3)。その値がレコード 6 の値リスト B にあれば、すなわち取得した値が値リスト B の値 V b と同じであればカウンタ N b を 1 つ増やし (ステップ S 2 4)、カウンタ N b の値がカウンタ N a の値より大きいかどうかを判定して (ステップ S 2 5)、カウンタ N b の値がカウンタ N a の値より大きければ、値リスト A と値リスト B とを入れ替える (ステップ S 2 6)。

【0079】

また、ステップ S 2 3 の処理において、その値がレコード 6 の値リスト B になれば、取得した値を値リスト B の値 V b に登録し、カウンタ N b の値を 1 にする (ステップ S 2 7)。

【0080】

実行の終了後、レコード 6 に記録された値がまとめられ、一次プロファイル情報 3 1 としてファイル出力される (ステップ S 1 0 5)。

【0081】

このようにして、実行中の手続きの変数に与えられる出現頻度が 5 0 % 以上になりうる値が一次プロファイル情報 3 1 として記憶される。

【0082】

次に、第 2 回目の翻訳処理を行うため、コンパイル処理装置 1 は、一時プロファイル読み込み部 1 3 および最終プロファイル処理部 1 4 を実施する指示を受けて起動される。

【0083】

まず、解析処理部 1 1 により、翻訳対象となる原始プログラム 4 を読み込み、ステップ S 1 0 1 の処理と同様、必要な構文解析、データ解析などを行い、中間形式データに変換する。その際に、プログラム内の全ての基本ブロックと変数に固有の番号を振る (ステップ S 1 0 6)。

【0084】

次に、一次プロファイル読み込み部 1 3 により、一次プロファイル情報 3 1 を読み込み、記憶されている 2 つの値をレコード 6 に設定する (ステップ S 1 0 7)。

【0085】

さらに、最終プロファイル処理部 1 4 により、プロファイル情報をとるよう指定した変数の 1 つ 1 つに対し、以下のような処理を行う (ステップ S 1 0 8)。

【0086】

まず、最終プロファイル処理部 1 4 は、中間形式データのデータ部分にその変数用のレコード 6 を用意する。なお、レコード 6 は、前記の統計処理において使用したのと同じデータ構造体である。そして、レコード 6 の基本ブロック番号欄と変数番号欄を設定し、一時プロファイル情報 3 1 から得た 2 つの値を、値リスト A および値リスト B にそれぞれ設定する。さらに、中間形式データ内のその変数の定義点の直後に、統計検証処理を行う命令列を挿入する。

【0087】

その後、目的プログラム生成部 1 7 により、中間形式データから目的プログラム 5 を生成する (ステップ S 1 0 9)。さらに、人手により、目的プログラム 5 を仮に実行する (ステップ S 1 1 0)。目的プログラム 5 の実行中、ステップ S 1 0 8 の処理で挿入された命令列により、以下のような統計検証処理が行われる。

【0088】

図 5 に、挿入された命令列による統計検証処理の流れを示す。

【0089】

まず、基本ブロックの実行回数カウンタを 1 つ増やす (ステップ S 3 0)。そして、実行中に手続き中の変数に与えられた値を取得し (ステップ S 3 1)、取得した値がレコード 6 の値リスト A にあるかどうかを判定する (ステップ S 3 2)。

【0090】

取得した値がレコード6の値リストAの値V aと同じであれば、カウンタN aを1つ増やす(ステップS 33)。

【0091】

一方、取得した値が値リストAになければ、その値がレコード6の値リストBにあるかどうかを判定する(ステップS 34)。

【0092】

そして、その値がレコード6の値リストBにあれば、すなわち取得した値が値リストBの値V bと同じであればカウンタN bを1つ増やす(ステップS 35)。

【0093】

実行の終了後、レコード6に記録された値がまとめられ、最終プロファイル情報32としてファイル出力される(ステップS 111)。

【0094】

このようにして、統計処理で抽出された2つの値の出現回数と、その値が与えられる変数を含む手続きの実行回数とが最終プロファイル情報32として記憶される。

【0095】

次に、第3回目の翻訳処理を行うため、コンパイル処理装置1は、最終プロファイル読み込み部15および最適化部16を実施する指示を受けて起動される。

【0096】

まず、解析処理部11により、翻訳対象となる原始プログラム4を読み込み、ステップS 101の処理と同様、必要な構文解析、データ解析などを行い、中間形式データに変換する。その際に、プログラム内の全ての基本ブロックと変数に固有の番号を振る(ステップS 112)。

【0097】

次に、最終プロファイル読み込み部15により、最終プロファイル情報32を読み込み、統計検証処理において設定された値リストAおよび値リストBの値について、それぞれの実行回数カウンタの値(N aまたはN b)を基本ブロック実行カウンタで割り、それぞれの値の出現頻度を求める(ステップS 113)。

【0098】

そして、値リストAまたは値リストBの値のうち出現頻度が50%以上の値があれば、最適化部16により、その値を利用した最適化を行う(ステップS 114)。そして、目的プログラム生成部17により、目的プログラム5を生成する(ステップS 115)。

【0099】

図6に、最終プロファイル情報32を用いた最適化の例を示す。

【0100】

図6(A)は、最適化前の原始プログラム4のある一部を示す。原始プログラム4の部分に記述されている関数f o oは、2つの整数引数を取り、その商を返す関数である。例えば、最終プロファイル読み込み部15により、最終プロファイル情報32を参照して、関数f o oの変数yのとりうる値が50%以上の確率で「17」であることが判明したとする。

【0101】

最適化部16は、関数f o oについて、高頻度に与えられる変数yの値=17の場合と、それ以外の値の場合とに分けて動作を行い、高頻度の値を固定化して扱い処理速度を高速化する最適化を行う。

【0102】

図6(B)は、最適化が施されて生成された目的プログラム5の図6(A)に示す部分を、可読性を確保して本発明の効果を明白にするために、原始プログラム4の記述形式で示したものである。図6(B)に示すように、この手続きにおいて頻度が50%以上となる変数yの値=17の場合の処理は、最適化前の処理より高速化される。

【0103】

このように、高い出現頻度（50%以上）で変数に与えられる値に対して特殊化するような最適化を行うことにより、その変数を含む手続き全体の高速化を図ることができる。結果的に、生成する目的プログラム5の高速化、効率化を図ることができる。

【0104】

以上、本発明をその実施の形態により説明したが、本発明はその主旨の範囲において種々の変形が可能であることは当然である。

【0105】

例えば、ステップS113の処理において、一次プロファイル情報31に基づいて設定したレコード6の値の出現頻度を求める割り算処理は、ステップS111の処理において、統計検証処理の結果と一緒に目的プログラム5に挿入して行い、最終プロファイル情報32に記録するようにしてもよい。この場合に、最終プロファイル処理部14は、上記の出現頻度を求める割り算処理を行う命令列を、中間形式データ中に埋め込む。

【0106】

また、統計処理および統計検証処理は、原始プログラム4に含まれる任意の変数を対象として行うようにしてもよい。この場合に、統計情報の対象とする変数に関する情報が、統計処理開始前にコンパイル処理装置1に与えられるようにする。なお、統計情報の対象とする変数は、ユーザが任意に設定できる。

【0107】

また、本発明は、コンピュータにより読み取られ実行される処理プログラムとして実施するものとして説明したが、本発明を実現する処理プログラムは、コンピュータが読み取り可能な、可搬媒体メモリ、半導体メモリ、ハードディスクなどの適当な記録媒体に格納することができ、これらの記録媒体に記録して提供され、または、通信インタフェースを介して種々の通信網を利用した送受信により提供されるものである。

【図面の簡単な説明】

【0108】

【図1】本発明の一実施例における構成例を示す図である。

【図2】本発明の処理フローを示す図である。

【図3】レコードの構成例を示す図である。

【図4】挿入された命令列による統計処理の流れを示す図である。

【図5】挿入された命令列による統計検証処理の流れを示す図である。

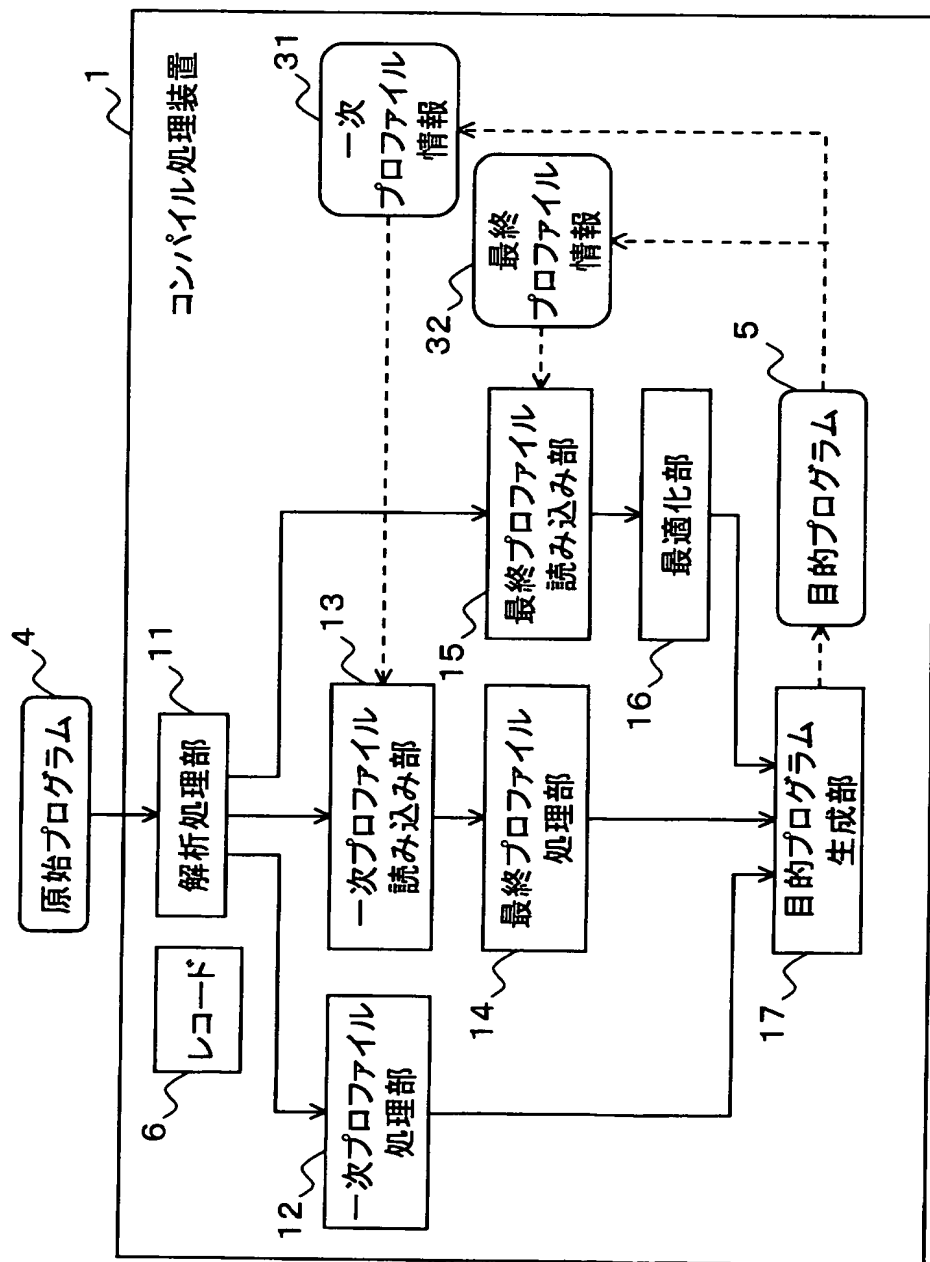
【図6】最終プロファイル情報を用いた最適化の例を示す図である。

【符号の説明】

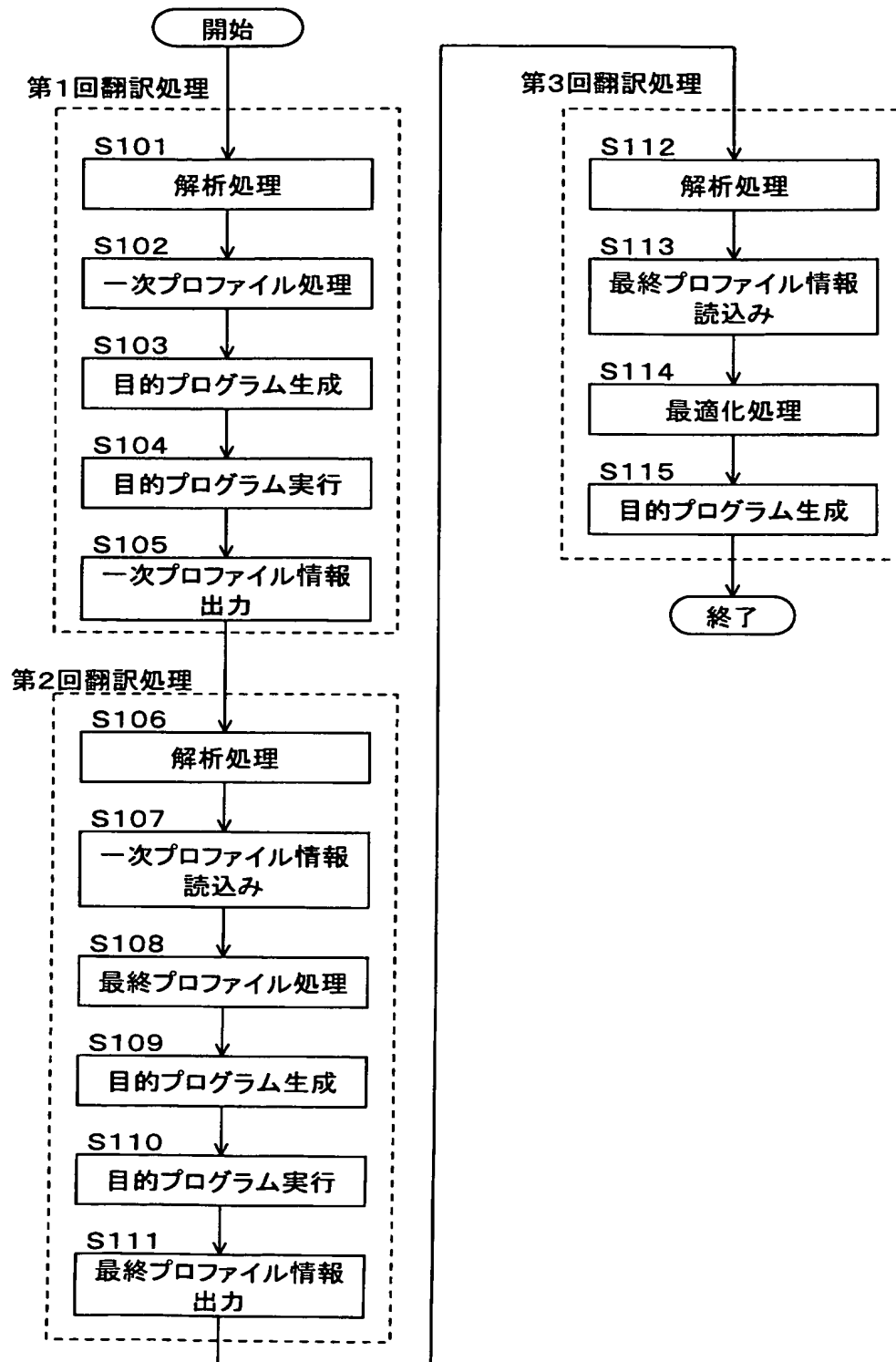
【0109】

- 1 コンパイル処理装置
- 11 解析処理部
- 12 一時プロファイル処理部
- 13 一時プロファイル読み込み部
- 14 最終プロファイル処理部
- 15 最終プロファイル読み込み部
- 16 最適化部
- 17 目的プログラム生成部
- 31 一時プロファイル情報
- 32 最終プロファイル情報
- 4 原始プログラム
- 5 目的プログラム
- 6 レコード

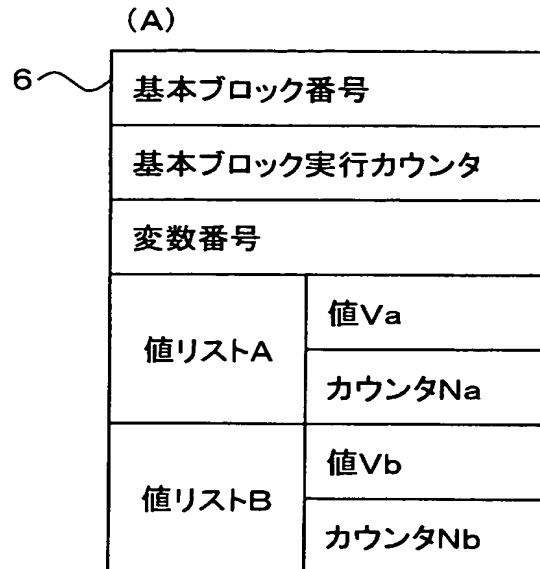
【書類名】 図面
【図 1】



【図 2】



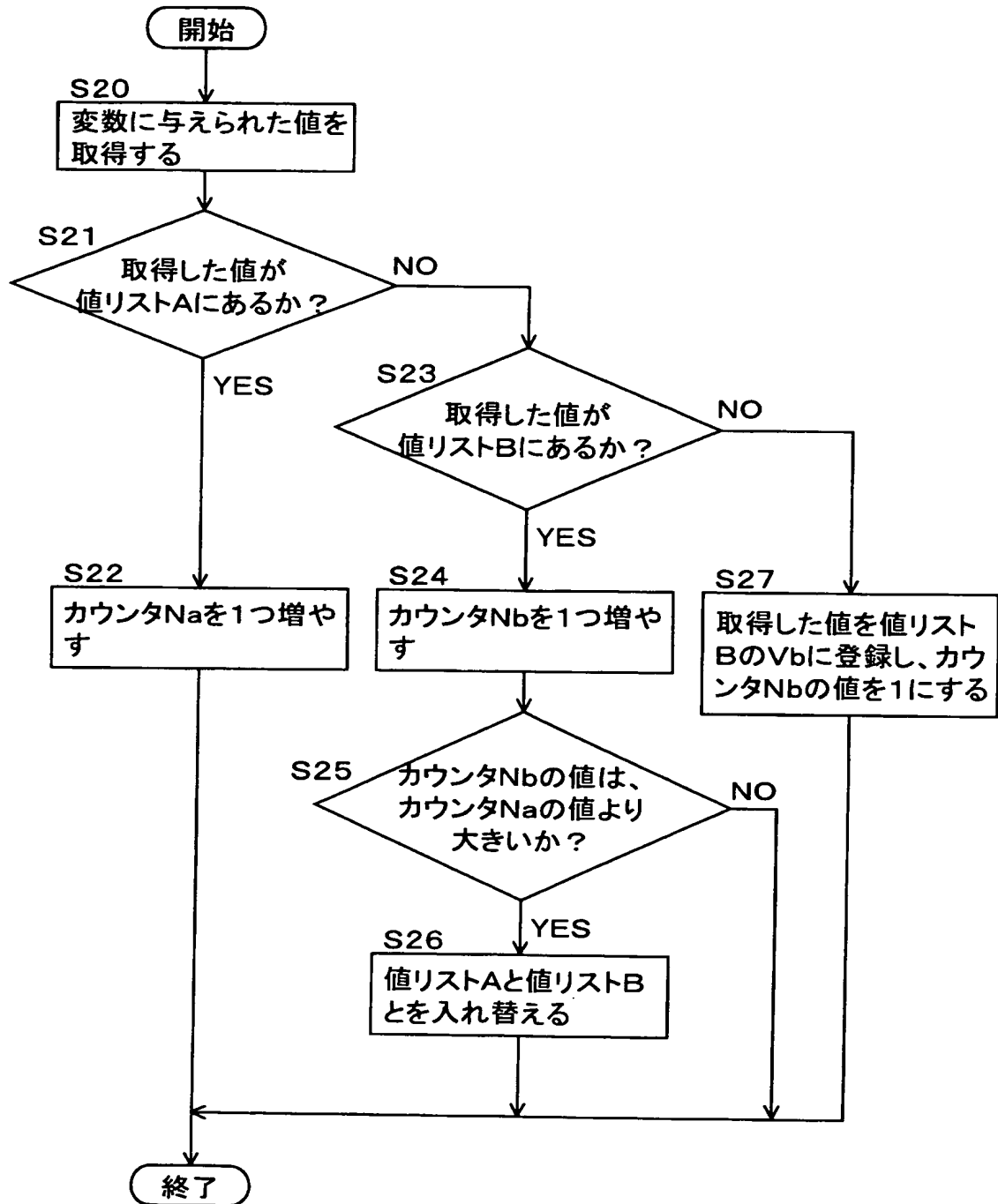
【図 3】



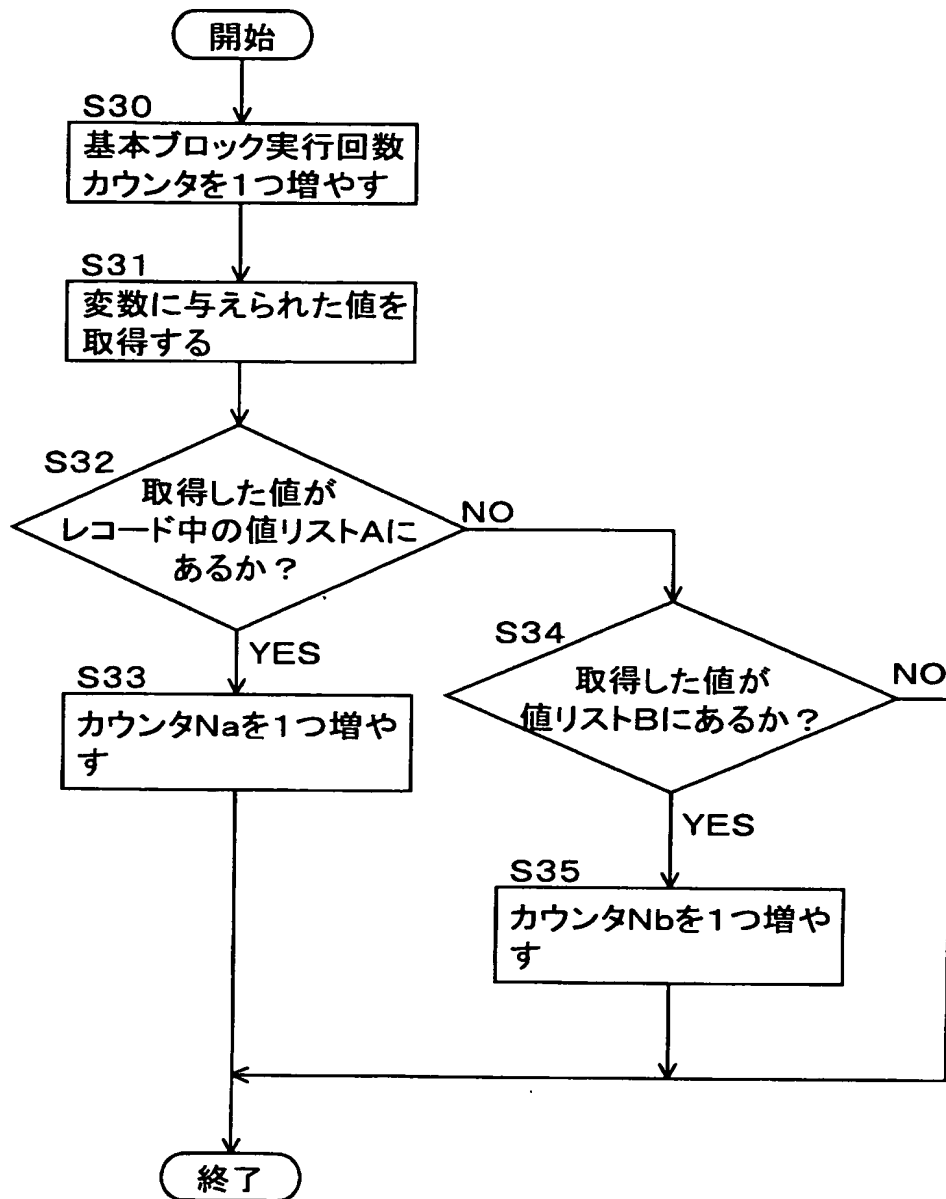
(B)

```
struct prof_record {  
    int bb_id;          /* コンパイラが基本ブロックにつける固有の番号 */  
    int bb_count;       /* 基本ブロックの実行回数カウンタ */  
    int var_id;         /* コンパイラが変数につける固有の番号 */  
    struct {            /* 値リスト */  
        int val;        /* 実行時に変数に与えられた値 */  
        int count;      /* 値が与えられた回数 */  
    } array [2]         /* 値リストの大きさは2である */  
};
```

【図 4】



【図 5】



【図 6】

(A) 最適化前

```
short foo (short x, short y)
{
    short d;

    d = x / y;
    return d;
}
```

(B) 最適化後

```
short foo (short x, short y)
{
    short d;

    if (y == 17) {
        return ((int)x * 3855) >> 16;
    } else {
        d = x / y;
        return d;
    }
}
```

【書類名】 要約書

【要約】

【課題】 プログラムのコンパイル処理において、実行中の手続き内の変数に高頻度で与えられる値のプロファイル情報を収集し、最適化に利用する。

【解決手段】 一次プロファイル処理部12は、解析処理部11で生成された中間形式データに、実行中の手続き内の変数に与えられる値で頻度が50% 以上になりうる値とその出現回数とを最大2組まで記録する処理を付加し、一次プロファイル情報31を得る。最終プロファイル処理部14は、中間形式データに、一次プロファイル情報31の2つ値の出現回数と手続きの実行回数を記録する処理を付加し、最終プロファイル情報32を得る。最適化部16は、最終プロファイル情報32をもとに変数に対し出現頻度が50% 以上の値を求め、この値に関する手続きを最適化し、目的プログラム生成部17は、目的プログラム5 を生成する。

【選択図】 図 1

特願 2 0 0 3 - 3 0 1 0 3 9

出 願 人 履 歴 情 報

識別番号

[0 0 0 0 0 5 2 2 3]

1. 変更年月日

1 9 9 6 年 3 月 2 6 日

[変更理由]

住所変更

住 所

神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号

氏 名

富士通株式会社